Join-Up-To(m): Improved Hyper-scalable Load Balancing

Grzegorz Kielanski[1], Tim Hellemans and Benny Van Houdt[1]

[1]Department of Computer Science, University of Antwerp,
Middelheimlaan 1, Antwerp, 2020, Belgium.

Contributing authors: grzegorz.kielanski@uantwerpen.be; timhellemanstim@gmail.com; benny.vanhoudt@uantwerpen.be;

Abstract

Various load balancing policies are known to achieve vanishing waiting times in the large-scale limit, that is, when the number of servers tends to infinity. These policies either require a communication overhead of one message per job or require job size information. Load balancing policies with an overhead below one message per job are called hyper-scalable policies. While these policies often have bounded queue length in the large-scale limit and work well when the overhead is somewhat below one, they show poor performance when the communication overhead becomes small, that is, the mean response time tends to infinity when the overhead tends to zero even at low loads. In this paper we introduce a hyper-scalable load balancing policy, called Join-Up-To($\boldsymbol{m}$), that remains effective even when the communication overhead tends to zero. To study its performance under general job size distributions, we make use of the "queue at the cavity" approach. We provide explicit results for the first two moments of the response time, the generating function of the queue length distribution and the Laplace transform of the response time. These results show that the mean response time only depends on the first two moments of the job size distribution.

Keywords: load balancing, hyper-scalable policies, cavity queue, vacation queues

MSC Classification: 60J28 , 60K25 , 68M20

1 Introduction

Load balancing plays a crucial role in any large-scale distributed system. If the dispatcher responsible for assigning jobs to servers has perfect knowledge of the servers that are idle, then it is intuitively clear that the waiting time of jobs vanishes as the number of servers tends to infinity when the system load is below 1. This can be achieved using a simple algorithm called Join-the-Idle-Queue, where a server informs the dispatcher whenever it becomes idle [1, 2] and incoming jobs are assigned to idle servers (if there is at least one idle server). It is also clear that in such case the communication overhead is one message per job. Vanishing waiting times have also been established in the so-called hyper-scalable regime [3, 4], that is, when the communication overhead is below one message per job, but this is not possible without information regarding the job sizes [5].

Hyper-scalable load balancers that do not require any job size information have been studied in [6] and [7]. Under the asynchronous policy introduced in [6] the dispatcher maintains an upper bound on the queue length of each server and assigns jobs in a greedy manner by selecting a server with the lowest upper bound. When a job is assigned to a server its upper bound is increased by one. The servers occasionally inform the dispatcher about their current state, that is, each server sends queue length information at some rate δ. When the dispatcher receives queue length information from a server, it updates its upper bound by setting it equal to the reported queue length. The pull policy presented in [7] works similarly, except that a server now sends its queue length information at rate δ_0 when it is idle and at rate δ_1 when it is busy. The value of δ_0 and δ_1 are set such that the overall rate of updates per server equals δ, that is, $\delta = (1-\lambda)\delta_0 + \lambda\delta_1$ when λ represents the load (as $1-\lambda$ is the fraction of the time that a server is idle).

These hyper-scalable policies do not have vanishing waits when fewer than one message per job is used, but instead have bounded queue lengths in the large-scale limit. However as the number of messages per job tends to zero, this upper bound as well as the mean response time tend to infinity. This means that these load balancers perform worse than simply assigning jobs to a random server (which requires no communication overhead at all) when the number of messages per job becomes small as demonstrated in Table 1. For instance, for a load of 0.8 we see that both the asynchronous policy of [6] and the pull policy of [7] perform far worse than random assignment when $\delta = 1/40$. For the pull policy we set the parameter $\delta_1 = 0$. Note that setting $\delta_1 > 0$ means that we must lower the rate δ_0. Now suppose we have two servers with an estimated queue length of 5, but their actual queue lengths equal 0 and 2, respectively, then it is better that the idle server sends an update. As such setting $\delta_1 = 0$ is expected to yield the best performance. The intuition for the poor performance for small δ in Table 1 is that due to the infrequent updates from the servers, the upper bounds maintained by the dispatcher become large (and very loose) and therefore the greedy nature of these schemes implies that a server receives many jobs in a very short time (that is, as a batch in the limit) whenever it sends an update on its queue length information.

In this paper we propose a new hyper-scalable load balancing scheme called Join-Up-To(m), abbreviated as JUT(m), that outperforms random assignment irrespective of the communication overhead (and coincides with random assignment when the communication overhead tends to zero). JUT(m) is also superior to the asynchronous schemes considered in [6] and [7], unless the communication overhead is fairly close to one message per job (see Table 1). Under the JUT(m) policy the dispatcher also maintains an upper bound on the queue length of each server and idle servers occasionally inform the dispatcher about their state (see Section 2 for details). Incoming jobs are assigned to a server with the lowest upper bound strictly below m, if such a server exists, and are assigned at random otherwise. When m is set equal to one, JUT(m) coincides with the Join-Idle-Queue scheme with sub-linear communication overhead [5].

It should be noted that [6] also presents synchronous schemes that have better performance than their asynchronous counterparts. However, the synchronous schemes require that all servers to update their queue length information simultaneously, which is problematic in a large-scale system as the dispatcher needs to process a huge number of updates at once in such case. Further note that the JUT(m) is also an asynchronous scheme for which one could in principle also devise a synchronous version. Given the limited practicality of any synchronous scheme, we did not explore such a version.

To understand the system behavior when the number of servers tends to infinity, we study the queue at the cavity for the Join-Up-To(m) policy (introduced in Section 3). We derive explicit expressions for the mean and the variance of the response time, for the parameter value of m that minimizes the mean response time and we derive expressions for the generating function and Laplace transform of the queue length distribution and response time distribution, respectively. These results show that the mean response time and optimal m value for the JUT(m) policy only depend on the first two moments of the jobs size distribution, while the variance of the response time also depends on the third moment. We also analytically invert the generating function of the queue length distribution in case of phase-type distributed job sizes and analytically invert the Laplace transform of the response time in case of exponential job sizes. Some discussion on the asymptotic exactness of the queue at the cavity approach is presented in Appendix A in case of bounded queues and exponential job sizes.

The paper is structured as follows. In Section 2 we describe the system under consideration and introduce the JUT(m) policy. The queue at the cavity approach, used to analyse the performance of JUT(m) in a large-scale setting, is discussed in Section 3. The analysis of the queue at the cavity is presented in Section 4, while Section 5 contains various numerical results. Finally, conclusions can be found in Section 6.

policy	δ	$\lambda = 0.5$	$\lambda = 0.8$	$\lambda = 0.95$
Random assignment	0	2	5	20
Asynchr. policy [6]	4/10	1.5932	3.1489	6.4542
	1/10	4.3563	10.6507	22.7754
	1/40	16.0860	41.3408	87.4644
Pull policy, $\delta_1 = 0$ [7]	4/10	1.0968	1.5	1.6540
	1/10	3	4.5	5.0671
	1/40	10.5	16.5	19.4944
JUT(m_{opt}), this paper	4/10	1.2170	1.5451	1.7287
	1/10	1.6886	2.7712	3.6549
	1/40	1.9069	3.9658	6.9149

Table 1 Mean response time of some existing hyperscalable policies for exponential job sizes with mean 1. All policies use on average $\delta/\lambda < 1$ messages per job.

2 System description and the JUT(m) policy

We consider a set of N homogeneous servers, each with its own infinite buffer, and a central dispatcher. Every server processes the jobs in its queue in First-Come-First-Served order. Jobs arrive at the dispatcher according to a Poisson process with rate $N\lambda$, with $0 < \lambda < 1$. The service requirements of a job have a general distribution G with mean one, i.e. $E[G] = 1$. For each server the dispatcher maintains an upper bound on its queue length. Henceforth, we refer to these upper bounds as "estimates".

The policy: The load balancing policy called Join-Up-To(m) (JUT(m)) relies on a single integer parameter m and operates as follows. When an arrival occurs and some servers have an estimate strictly below m, then the dispatcher assigns the job to a server with lowest estimate among all such servers (with ties broken uniformly at random). Otherwise, if all estimates are at least m, the dispatcher assigns the job to a random server. Whenever the dispatcher assigns a job to a server, it increases this estimate by one. The estimate of a server can also be reset to zero. This happens when an idle server informs the dispatcher that its queue is empty. In order to have an average of $\delta/\lambda < 1$ such messages per arrival, idle servers inform the dispatcher about their state at rate $\delta_0 = \delta/(1-\lambda)$ as $1-\lambda$ is the fraction of time that a server is idle.

The parameter m: When $m > \lfloor \lambda/\delta \rfloor$ the performance of the JUT(m) policy coincides with the pull policy in [7] (with $\delta_1 = 0$) when the number of servers tends to infinity (as the dispatcher never runs out of servers with an estimate strictly below m). Recall that this policy becomes inferior to random assignment for δ small enough. We therefore focus on JUT(m) with $m \leq \lfloor \lambda/\delta \rfloor$. In Corollary 6 we present a simple explicit expression that depends only on λ, δ and $E[G^2]$ for the value of m that minimizes the mean response time (for the queue at the cavity with $E[G] = 1$).

3 Queue at the cavity approach

As the system of N servers is hard to analyze directly and simulation experiments do not provide closed form expressions and become very time consuming for large N, we make use of the so-called "queue at the cavity approach" [8]. The basic idea of this approach is to focus on the evolution of a single server and to assume that all other servers have independent and identically distributed queue lengths. In some particular cases the queue at the cavity method was proven to yield exact results as the number of servers tends to infinity (see [8, 9]). The system that is closest to ours for which such a proof was established is [10] which was limited to exponential job sizes. In Appendix A we prove that the stochastic system with N servers converges to the set of solutions of a differential inclusion as N tends to infinity over finite time scales in case of exponential service times and finite buffers. We also identify the missing piece that is required to extend this convergence result to the stationary regime. In this section we limit ourselves to presenting a set of simulation results which suggest that the queue at the cavity also yields exact results as N tends to infinity in our setting. We start with a number of randomly selected cases and increasing N. Afterwards we fix N at 1000 servers and vary the different system parameters.

In Table 2, we compare the relative error of Corollary 4 for the queue at the cavity with the simulated mean response time, for $N \in \left\{10^2, 10^3, 10^4, 10^5\right\}$, based on 20 runs. Each run contains $1000N$ arrivals and has a warm-up period of 10%. We consider different randomly chosen values of λ, δ, m and different job size distributions. The job size distributions considered in Table 2 are exponential, Erlang, hyperexponential (HypExp), hyper-Erlang (HypErl) and truncated Pareto, all with mean 1. The hyperexponential distribution of order 2 is described using the shape parameter f and the squared coefficient of variation SCV [11]. The hyper-Erlang distribution HypErl(k, ℓ) is such that jobs are Erlang-k with probability p and Erlang-ℓ otherwise. The truncated Pareto distribution is characterized by three values: α, L and U, with $0 < L < U < +\infty$. Its CDF is given by $F(x) = (1-(L/x)^\alpha)/(1-(L/U)^\alpha)$ for $x \in [L, U]$. The value of α is called the shape parameter, while L and U respectively denote the lower and upper bound of the support of the distribution. Note, that we require that jobs have mean 1. Hence, if X has a Pareto distribution with parameters α, L and U, we instead work with the random variable $X/E[X]$ and denote $X/E[X] \sim$ Pareto$(\alpha, [L, U])$.

The simulation results presented in Table 2 are a random subset of simulations which we performed. When $N \geq 1000$ the error always stays below 1.5%. Further, in all cases the simulated mean response time seems to be $O(1/N)$ accurate in function of N, similar to the results in [12]. We now set N equal to 1000 and vary the different system parameters to demonstrate that the approximation is highly accurate in a variety of settings. In Figure 1 we consider 3 job size distributions: exponential, hyper-exponential and truncated Pareto. In the first plot we vary m with $\lambda = 0.95$ and $\delta = 0.1$, in the second plot the impact of changing λ is shown for $m = 5$ and $\delta = 0.1$ and in the third plot δ varies with $m = 5$ and $\lambda = 0.9$. The red lines are the results obtained using the queue at the cavity, the black curves are simulation results with confidence intervals added. The simulation results are based on 50 runs with 10^7 arrivals

settings	N	sim. ± conf.	rel.err.%
Exponential	100	5.3145 ± 7.28e-03	2.4216
$\lambda = 0.8$	1000	5.4347 ± 1.75e-03	0.2147
$\delta = 0.05$	10000	5.4450 ± 5.95e-04	0.0268
$m = 10$	100000	5.4463 ± 1.38e-04	0.0021
	∞	5.4464	0
HypExp(2)	100	11.7581 ± 5.55e-01	11.6331
$f = 1/2, SCV = 15$	1000	10.6572 ± 2.14e-01	1.1811
$\lambda = 0.95$	10000	10.5240 ± 4.34e-02	0.0836
$\delta = 0.05$	100000	10.5349 ± 1.77e-02	0.0195
$m = 15$	∞	10.5328	0
Erlang(10)	100	3.4231 ± 9.73e-03	1.7598
$\lambda = 0.95$	1000	3.3728 ± 2.69e-03	0.2661
$\delta = 0.1$	10000	3.3651 ± 1.17e-03	0.0378
$m = 5$	100000	3.3640 ± 3.58e-04	0.0046
	∞	3.3639	0
HypErl(3,7)	100	2.0574 ± 3.32e-03	2.0553
$p = 0.85$	1000	2.0960 ± 6.11e-04	0.2186
$\lambda = 0.9$	10000	2.1005 ± 2.61e-04	0.0053
$\delta = 0.05$	100000	2.1006 ± 7.44e-05	0.0001
$m = 7$	∞	2.1006	0
Pareto(3, $[1, 50]$)	100	4.4010 ± 1.09e-02	0.2114
$\lambda = 0.9$	1000	4.3940 ± 2.13e-03	0.0522
$\delta = 0.05$	10000	4.3914 ± 8.19e-04	0.0088
$m = 7$	100000	4.3918 ± 3.67e-04	0.0002
	∞	4.3917	0

Table 2 Relative error of the simulated mean response time for the JUT(m) strategy based on 20 runs.

each. Careful examination of the plots shows that the approximation becomes somewhat less accurate as the job size variability and load increases, which is in agreement with intuition. The impact of m and δ on the accuracy appears to be less pronounced.

4 Analysis of the queue at the cavity

4.1 General job sizes

The queue at the cavity for the JUT(m) policy is defined as an M/G/1 queue with arrival rate $\tilde{\lambda} = \lambda - \delta m$, except that when the queue is empty there are also batch arrivals of size m that occur at rate $\delta_0 = \delta/(1-\lambda)$. The intuition behind this queue at the cavity is that in the large-scale limit any idle server that informs the dispatcher about its state (which occurs at rate δ_0) will immediately receive a batch of m jobs. As the overall rate of such messages is δ, this implies that a fraction $\delta m/\lambda$ of the jobs is assigned in this manner. The remaining fraction $1 - \delta m/\lambda$ of the jobs is assigned at random and corresponds to the arrivals at rate $\tilde{\lambda} = \lambda - \delta m$. Recall that we assume that $m < \lambda/\delta$, such that $\tilde{\lambda} > 0$. As m is an integer, we have $m \leq \lfloor \lambda/\delta \rfloor$.

The fact that the queue length of any idle server that updates its queue length information immediately jumps to m for the cavity queue is due to the

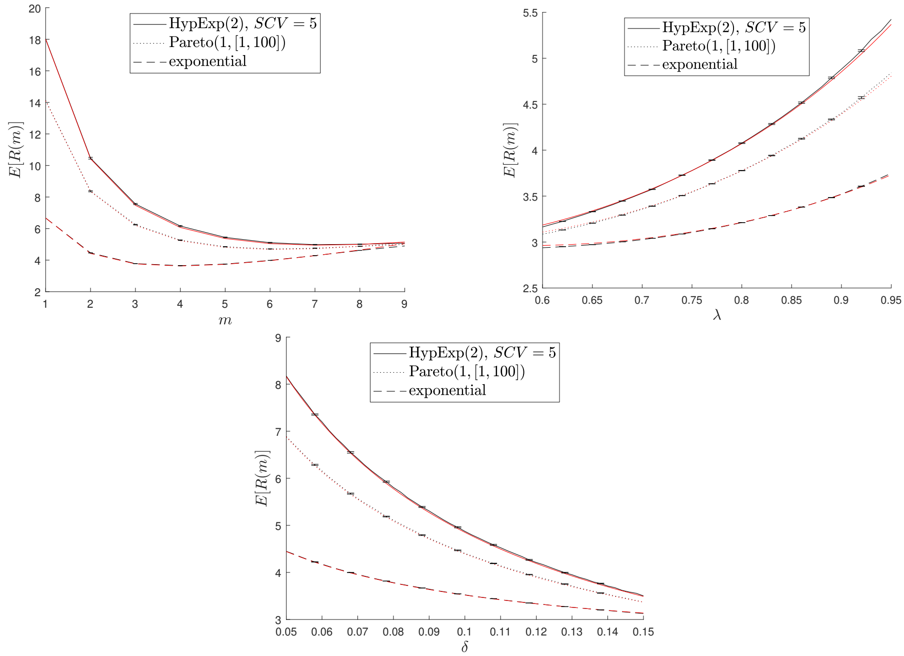

Fig. 1 Comparison of the mean response time obtained by simulation with $N = 1000$ servers (black) and the queue at the cavity (red) for different scenarios with varying m, λ and δ.

greedy nature of the JUT(m) algorithm whenever there are servers available with a queue length estimate strictly below m. Note that the same phenomena also occurs for the cavity queue of the asynchronous policy in [6] (see Section 4 in [7]) which is also in agreement with the fixed point analysis in [6] that suggests that a server has an estimated queue length of m^* or $m^* + 1$ with probability one.

Let π_i^a, π_i^d and π_i be the steady state probability that there are i jobs in the cavity queue at arrival, departure and at a random time, respectively. Let $\pi^a(z), \pi^d(z)$ and $\pi(z)$ be the associated generating functions. Note that if a job is the k-th job of a batch of size m, then it sees $k-1$ jobs at arrival time. It is well known and easy to see that $\pi^a(z) = \pi^d(z)$. The next theorem relates $\pi(z)$ with $\pi^a(z)$.

Theorem 1 *The generating function $\pi(z)$ can be written as*

$$\pi(z) = \frac{\lambda}{\tilde{\lambda}} \pi^a(z) - \frac{\delta}{\tilde{\lambda}} \frac{1 - z^m}{1 - z}. \tag{1}$$

Proof Consider a tagged arrival (that potentially arrives in a batch of size m). In order for the tagged arrival to observe $i < m$ jobs upon arrival there are two options. First, the tagged job arrives when the queue length equals exactly i with probability

$\tilde{\lambda}\pi_i/\lambda$. Second the tagged arrival is the $i+1$-st arrival in a batch arrival of size m. This occurs with probability $\delta_0\pi_0/\lambda$ as $\delta_0\pi_0$ is the rate at which batches of size m arrive and each batch contains exactly one job in position $i+1$. Hence, we have the following relationship between the probabilities π_i^a and π_i:

$$\pi_i^a = \left(\tilde{\lambda}\pi_i + \delta_0\pi_0\right)/\lambda,$$

for $0 \leq i < m$. Further, as batch arrivals only occur when the queue is empty, we have

$$\pi_i^a = \tilde{\lambda}\pi_i/\lambda,$$

for $i \geq m$. Hence,

$$\pi^a(z) = \frac{\delta}{\lambda}\sum_{i=0}^{m-1} z^i + \frac{\tilde{\lambda}}{\lambda}\pi(z) = \frac{\delta}{\lambda}\frac{1-z^m}{1-z} + \frac{\tilde{\lambda}}{\lambda}\pi(z),$$

and therefore (1) holds. □

Recall that for an $M/G/1$ queue with arrival rate $\tilde{\lambda}$ and mean job size 1, the generating function of the queue length is given by the Pollaczek-Khinchin formula [13, (5.32)]

$$\xi(z) = \frac{(1-\tilde{\lambda})(1-z)G^*(\tilde{\lambda}-\tilde{\lambda}z)}{G^*(\tilde{\lambda}-\tilde{\lambda}z)-z}, \tag{2}$$

where $G^*(s)$ is the Laplace-Stieltjes transform of the job size distribution.

Theorem 2 *The generating function $\pi^a(z)$ can be written as*

$$\pi^a(z) = \frac{1-\alpha(z)}{\alpha'(1)(1-z)}\xi(z), \tag{3}$$

where $\xi(z)$ is given by (2) and

$$\alpha(z) = \frac{\tilde{\lambda}}{\tilde{\lambda}+\delta_0}z + \frac{\delta_0}{\tilde{\lambda}+\delta_0}z^m.$$

Note that $\alpha'(1) = (\tilde{\lambda}+m\delta_0)/(\tilde{\lambda}+\delta_0)$.

Proof Consider an M/G/1 queue with arrival rate $\tilde{\lambda}$ where the server starts a vacation each time the queue becomes empty. The vacation ends with probability $\tilde{\lambda}/(\tilde{\lambda}+\delta_0)$ when an arrival occurs or ends when the m-th arrival occurs otherwise. The queue length distribution of this vacation queue is the same at arrival, departure and at random times (due to PASTA) and its generating function $\phi(z)$ obeys the well known decomposition result for vacation queues [14, 15], that is,

$$\phi(z) = \frac{1-\alpha(z)}{\alpha'(1)(1-z)}\xi(z),$$

where $\xi(z)$ is the generating function of the queue length of a standard M/G/1 queue with arrival rate $\tilde{\lambda}$ and $\alpha(z)$ is the generating function of the number of arrivals during a vacation.

The proof completes by noting that the queue length distribution at departure times in the queue at the cavity $\pi^d(z)$ and at departure times in the vacation queue $\phi(z)$ are the same, while $\pi^a(z) = \pi^d(z)$. □

Corollary 3 *The generating function $\pi(z)$ is given by*

$$\pi(z) = \frac{\lambda}{\tilde{\lambda}}\beta(z)\xi(z) - \frac{\delta}{\tilde{\lambda}}\frac{1-z^m}{1-z}, \tag{4}$$

with $\beta(z) = (1-\alpha(z))/(\alpha'(1)(1-z))$.

Proof This is immediate from the previous two theorems. □

Note that $\beta(z)$ is the generating function of the number of customers that arrive during a vacation period after the arrival of the random customer during a vacation [14]. Using this interpretation we note that

$$\beta(z) = \frac{\tilde{\lambda}}{\delta_0 m + \tilde{\lambda}} + \frac{\delta}{\lambda(1-\tilde{\lambda})}\sum_{i=0}^{m-1} z^i, \tag{5}$$

as $\delta_0 m/(\delta_0 m + \tilde{\lambda}) = \delta m/(\lambda(1-\tilde{\lambda}))$ is the probability that the random arrival is part of a vacation with m arrivals and its position is uniform within these m arrivals.

Corollary 4 *The mean response time $E[R(m)]$ in the queue at the cavity is given by*

$$E[R(m)] = 1 + \frac{\tilde{\lambda}E[G^2]}{2(1-\tilde{\lambda})} + \frac{\delta}{\lambda}\frac{m(m-1)}{2(1-\tilde{\lambda})}. \tag{6}$$

In particular, we have

$$\lim_{\lambda\to 1^-} E[R(m)] = \frac{1-\delta m}{2\delta m}E[G^2] + \frac{m+1}{2}. \tag{7}$$

Proof Due to Little, we have $E[R(m)] = \pi'(1)/\lambda$. Using (4) we have

$$\frac{\pi'(1)}{\lambda} = \frac{\beta'(1)}{\tilde{\lambda}} + \frac{\xi'(1)}{\tilde{\lambda}} - \frac{\delta}{\tilde{\lambda}}\frac{m(m-1)}{2}\frac{1}{\lambda},$$

where $\xi'(1)/\tilde{\lambda}$ is the mean response time in a standard M/G/1 queue with arrival rate $\tilde{\lambda}$, which equals $1 + \tilde{\lambda}E[G^2]/(2(1-\tilde{\lambda}))$ as $E[G] = 1$. The first claim therefore follows by verifying that

$$\beta'(1) = \frac{\delta m(m-1)}{2\lambda(1-\tilde{\lambda})},$$

which is immediate from (5). The second claim follows immediately from (6) as $\tilde{\lambda}$ converges to $1-\delta m$. □

Remark: As long as $m > 0$ and $\delta > 0$, the mean response time remains bounded when λ tends to one, in contrast to an ordinary M/G/1 queue. The mean response time of JUT(m) converges to that of random assignment when δ tends to zero, which is an improvement over the policies in [6, 7] where the mean response time tends to infinity as δ tends to zero for any $\lambda < 1$.

Theorem 5 *The Laplace transform $R^*(s)$ of the response time distribution of the queue at the cavity can be expressed as*

$$R^*(s) = \frac{\tilde{\lambda}}{\lambda} Y^*(s)\pi(G^*(s)) - \frac{\tilde{\lambda}(1-\lambda)}{\lambda}(Y^*(s) - G^*(s)) \\ + \frac{\delta}{\lambda}\left(\frac{1-G^*(s)^{m+1}}{1-G^*(s)} - 1\right), \tag{8}$$

where $G^*(s)$ and $Y^*(s)$ are the Laplace transforms of the service time and residual service time, respectively. It is also well known that $Y^*(s) = (1-G^*(s))/s$ as $E[G] = 1$.

Proof The arrivals that occur at rate $\tilde{\lambda}$ arrive at random points in time, therefore such an arrival sees a workload of one residual service time and $i-1$ service times with probability π_i, for $i > 0$ and no workload with probability π_0. For a tagged arrival that occurs in a batch of size m the workload observed upon arrival is equal to the service time of the jobs that are part of the same batch and ahead of the tagged arrival. This implies

$$\begin{aligned} R^*(s) &= \frac{\tilde{\lambda}}{\lambda}\left(\sum_{i=1}^{\infty} \pi_i Y^*(s) G^*(s)^i + \pi_0 G^*(s)\right) \\ &\quad + \frac{\delta m}{\lambda} \sum_{i=0}^{m-1} \frac{1}{m} G^*(s)^{i+1} \\ &= \frac{\tilde{\lambda}}{\lambda}\left(Y^*(s)(\pi(G^*(s)) - \pi_0) + \pi_0 G^*(s)\right) \\ &\quad + \frac{\delta}{\lambda}\left(\frac{1-G^*(s)^{m+1}}{1-G^*(s)} - 1\right). \end{aligned}$$

Equation (8) then follows as $\pi_0 = 1-\lambda$. □

Remark: We can also retrieve (6) using (8) by making use of the fact that $E[R(m)] = -R^{*\prime}(0)$. More specifically, we can make use of the equalities $G^{*\prime}(0) = -E[G] = -1$ and $-Y^{*\prime}(0) = E[Y] = E[G^2]/2$ to find that

$$\begin{aligned} E[R(m)] &= -R^{*\prime}(0) \\ &= -\frac{\tilde{\lambda}}{\lambda}\left(Y^{*\prime}(0) - \pi'(1) - (1-\lambda)(Y^{*\prime}(0)+1)\right) \\ &\quad + \frac{\delta}{\lambda}\frac{m(m+1)}{2} \\ &= \tilde{\lambda}\frac{\pi'(1)}{\lambda} - \tilde{\lambda}Y^{*\prime}(0) + \frac{\tilde{\lambda}(1-\lambda)}{\lambda} + \frac{\delta m}{\lambda} + \frac{\delta}{\lambda}\frac{m(m-1)}{2} \\ &= \tilde{\lambda}E[R(m)] + \frac{\tilde{\lambda}E[G^2]}{2} + (1-\tilde{\lambda}) + \frac{\delta}{\lambda}\frac{m(m-1)}{2}, \end{aligned}$$

as $\delta m + \tilde{\lambda} = \lambda$. Similarly, we can derive an explicit expression for the second moment of the response time $E[R(m)^2]$, see Appendix.

Theorem 6 *The mean response time of the queue at the cavity $E[R(m)]$ is minimized by setting m equal to $m_{opt} = \min(\hat{m}, \lfloor \lambda/\delta \rfloor)$ with*

$$\hat{m} = \left\lceil \sqrt{\left(\frac{1}{2} + \frac{1-\lambda}{\delta}\right)^2 + \frac{\lambda}{\delta} E[G^2]} - \left(\frac{1}{2} + \frac{1-\lambda}{\delta}\right) \right\rceil. \tag{9}$$

Proof Using (6) we get that

$$0 = \frac{\partial E[R(m)]}{\partial m} = \frac{\delta(\delta m^2 + (2m-1)(1-\lambda) - \lambda E[G^2])}{2\lambda(1-\lambda+\delta m)^2}. \tag{10}$$

This equation has a unique positive root given by

$$m^* = \frac{\sqrt{(1-\lambda)^2 + \delta\left[1 + \lambda\left(E[G^2]-1\right)\right]} - (1-\lambda)}{\delta},$$

as $1 + \lambda(E[G^2]-1) > 0$. One readily verifies that

$$\frac{\partial^2 E[R(m)]}{\partial m^2} = \frac{\delta((1-\lambda)\delta + (1-\lambda)^2 + \lambda\delta E[G^2])}{\lambda(1-\lambda+\delta m)^3} \geq 0.$$

for $m \geq 0$. Therefore $E[R(m)]$ is convex in m on $[0,\infty)$ and m^* is the minimum of $E[R(m)]$. However m^* is typically not an integer.

The integer value that minimizes $E[R(m)]$ is found by defining $\Delta_R(m) = E[R(m+1)] - E[R(m)]$ and taking the ceil of its unique positive root. By further using (6), one easily checks that

$$\Delta_R(m) = \frac{\delta m}{N\lambda}\left(1 - \lambda + \delta\frac{(m+1)}{2}\right) - \frac{\delta}{N} \cdot \frac{E[G^2]}{2},$$

where $N = (1-\lambda)^2 + (1-\lambda)\delta(2m+1) + \delta^2 m(m+1) > 0$. We have that $\Delta_R(m) = 0$ if and only if

$$\frac{m}{\lambda}\left(1 - \lambda + \frac{\delta(m+1)}{2}\right) - \frac{E[G^2]}{2} = 0,$$

which has a unique positive root given by

$$m = \frac{-\frac{\delta}{2\lambda} - \frac{1-\lambda}{\lambda} + \sqrt{\left(\frac{\delta}{2\lambda} + \frac{1-\lambda}{\lambda}\right)^2 + \frac{\delta}{\lambda}E[G^2]}}{\delta/\lambda}. \tag{11}$$

As $\tilde{\lambda} > 0$, we have $m \leq \lfloor \lambda/\delta \rfloor$ and the result follows by the convexity of $E[R(m)]$. □

Remark: The optimal value m_{opt} is decreasing in δ and increasing in both λ and $E[G^2]$. The next Corollary therefore implies that $m_{opt} \leq \lceil \lambda E[G^2]/(2(1-\lambda)) \rceil$.

Corollary 7 *The optimal value of m when $\delta \to 0^+$ is given by*

$$m_{opt}^{\delta \to 0^+} = \left\lceil \frac{1}{2}\frac{\lambda}{1-\lambda}E[G^2] \right\rceil.$$

The optimal value of m when $\lambda \to 1^-$ is given by

$$m_{opt}^{\lambda \to 1^-} = \min\left(\left\lceil \sqrt{\frac{1}{4} + \frac{E[G^2]}{\delta}} - \frac{1}{2} \right\rceil, \left\lfloor \frac{1}{\delta} \right\rfloor\right).$$

Proof The first claim follows by application of l'Hôpital's rule on (9), the second is immediate. □

Remark: We have

$$\begin{aligned}\frac{\partial E[R(m)]}{\partial \delta} &= -\frac{m\left(1-\lambda+\lambda E[G^2]-(1-\lambda)m\right)}{2\lambda(1-\tilde{\lambda})^2} \\ &= -\frac{(1-\lambda)m\left(1+\frac{\lambda}{1-\lambda}E[G^2]-m\right)}{2\lambda(1-\tilde{\lambda})^2}. \end{aligned} \tag{12}$$

We distinguish three possibilities:

1. If $m > 1+\frac{\lambda}{1-\lambda}E[G^2]$, then (12) is greater than 0, hence increasing the value of δ increases the mean response time. In this case, having $\delta = 0$ works the best. If $\delta = 0$, the queue at the cavity becomes a standard M/G/1 queue with arrival rate λ and (6) simplifies to $1+\frac{\lambda}{1-\lambda}\frac{E[G^2]}{2}$.
2. If $m = 1+\frac{\lambda}{1-\lambda}E[G^2]$, then (12) is 0, which implies that in this case $E[R(m)]$ is independent of δ. In fact, substituting $m = 1+\frac{\lambda}{1-\lambda}E[G^2]$ into (6) gives $E[R(m)] = 1+\frac{\lambda}{1-\lambda}\frac{E[G^2]}{2}$.
3. If $m < 1+\frac{\lambda}{1-\lambda}E[G^2]$, then (12) is smaller than 0 and the proposed policy works better than random assignment for any $\delta > 0$.

Note that when $m = 1$ or $m = m_{opt}$ (due to Corollary 7), case 3) applies and our policy improves upon random assignment.

4.2 Phase-type distributed job sizes

In this section we provide an explicit formula for the queue length distribution in case of phase type distributed jobs, meaning we present an explicit formula for the probabilities π_k appearing in the generating function $\pi(z) = \sum_k \pi_k z^k$. Recall that a phase type distribution with n_p phases can be characterized by a couple (α, S), where α is a row vector of length n_p and is called the initial distribution vector, as α_i is the probability that the distribution starts in phase i; and where S is a $n_p \times n_p$ matrix that records the rates of phase changes.

Theorem 8 *Suppose G is PH(α, S) distributed (with mean 1). Then, for $k = 1, \ldots, m$:*

$$\pi_k = \left(1-\lambda+\frac{\delta}{\tilde{\lambda}}\right)\alpha R^k 1_{n_p} + \frac{\delta}{\tilde{\lambda}}\alpha(I-R)^{-1}\left(R-R^k\right)1_{n_p}, \tag{13}$$

and for $k > m$:

$$\pi_k = \left[\left(1-\lambda+\frac{\delta}{\tilde{\lambda}}\right)\alpha + \frac{\delta}{\tilde{\lambda}}\alpha(I-R)^{-1}\left(R^{1-m}-I\right)\right]R^k 1_{n_p}, \tag{14}$$

where $R = -\tilde{\lambda}(S-\tilde{\lambda}I+\tilde{\lambda}1_{n_p}\alpha)^{-1}$ and where 1_{n_p} is a column vector of ones of height n_p.

Proof By using [16, Theorem 3.2.1], we get

$$\xi(z) = (1-\tilde{\lambda}) \sum_{k=0}^{\infty} \alpha R^k 1_{n_p} z^k.$$

Therefore, by (4), (5) and the fact that $\lambda(1-\tilde{\lambda})/(\delta_0 m + \tilde{\lambda}) = 1 - \lambda$, we have

$$\begin{aligned}\pi(z) &= (1-\lambda) \sum_{k=0}^{\infty} \alpha R^k 1_{n_p} z^k \\ &\quad + \frac{\delta}{\tilde{\lambda}} \sum_{i=0}^{m-1} z^i \sum_{k=0}^{\infty} \alpha R^k 1_{n_p} z^k - \frac{\delta}{\tilde{\lambda}} \sum_{i=0}^{m-1} z^i \\ &= (1-\lambda) \sum_{k=0}^{\infty} \alpha R^k 1_{n_p} z^k + \frac{\delta}{\tilde{\lambda}} \sum_{i=0}^{m-1} z^i \sum_{k=1}^{\infty} \alpha R^k 1_{n_p} z^k \\ &= (1-\lambda) \sum_{k=0}^{\infty} \alpha R^k 1_{n_p} z^k + \frac{\delta}{\tilde{\lambda}} \sum_{k=1}^{\infty} \alpha R^k 1_{n_p} z^k \\ &\quad + \frac{\delta}{\tilde{\lambda}} \sum_{i=1}^{m-1} z^i \sum_{k=1}^{\infty} \alpha R^k 1_{n_p} z^k.\end{aligned}$$

On the other hand, by using (13)-(14) and $\pi_0 = 1 - \lambda$, we get

$$\begin{aligned}\sum_{k=0}^{\infty} \pi_k z^k &= (1-\lambda) \sum_{k=0}^{\infty} \alpha R^k 1_{n_p} z^k \\ &\quad + \frac{\delta}{\tilde{\lambda}} \sum_{k=1}^{\infty} \alpha R^k 1_{n_p} z^k + \frac{\delta}{\tilde{\lambda}} \sum_{k=1}^{m} \alpha (I-R)^{-1} \left(R - R^k\right) 1_{n_p} z^k \\ &\quad + \frac{\delta}{\tilde{\lambda}} \sum_{k=m+1}^{\infty} \alpha (I-R)^{-1} \left(R^{k-m+1} - R^k\right) 1_{n_p} z^k.\end{aligned}$$

Hence, it suffices to show that

$$\begin{aligned}\sum_{i=1}^{m-1} z^i \sum_{k=1}^{\infty} R^k z^k &= \sum_{k=1}^{m} (I-R)^{-1} \left(R - R^k\right) z^k \\ &\quad + \sum_{k=m+1}^{\infty} (I-R)^{-1} \left(R^{k-m+1} - R^k\right) z^k. \end{aligned} \tag{15}$$

The RHS of (15) equals

$$\sum_{k=2}^{m} z^k \sum_{\ell=1}^{k-1} R^\ell + \sum_{k=m+1}^{\infty} z^k \sum_{\ell=k-m+1}^{k-1} R^\ell, \tag{16}$$

while the LHS is equal to

$$\sum_{\ell=1}^{m-1} z^\ell \sum_{k=1}^{m-\ell} R^k z^k + \sum_{\ell=1}^{m-1} z^\ell \sum_{k=m+1-\ell}^{\infty} R^k z^k, \tag{17}$$

where the first sum of (17) consists of all terms with the exponent of z smaller than or equal to m and the second with exponents greater than m. The first sum of (17) equals

$$\sum_{\ell=1}^{m-1} \sum_{k=\ell+1}^{m} R^{k-\ell} z^k = \sum_{\ell=2}^{m} \sum_{k=\ell}^{m} R^{k-\ell+1} z^k$$

$$= \sum_{k=2}^{m} z^k \sum_{\ell=2}^{k} R^{k-\ell+1} = \sum_{k=2}^{m} z^k \sum_{\ell=1}^{k-1} R^{\ell},$$

which is the first sum of (16). Proceeding similarly with the second sum of (17), we get that it equals

$$\sum_{\ell=1}^{m-1} \sum_{k=m+1}^{\infty} R^{k-\ell} z^k = \sum_{k=m+1}^{\infty} z^k \sum_{\ell=k-m+1}^{k-1} R^{\ell},$$

which is the second sum of (16). This finishes the proof. □

Remark: Let $\Pi_{k,i}$ be the probability that the queue at the cavity contains k jobs and the job in service is in phase i. Denote $\Pi_k = \left[\Pi_{k,1}, \Pi_{k,2}, \ldots, \Pi_{k,n_p}\right]$, for $k > 0$. With some additional effort one can generalize the previous theorem and show that for $k = 1, \ldots, m$:

$$\Pi_k = \left(1 - \lambda + \frac{\delta}{\tilde{\lambda}}\right) \alpha R^k + \frac{\delta}{\tilde{\lambda}} \alpha (I - R)^{-1} \left(R - R^k\right),$$

and for $k > m$:

$$\Pi_k = \left(1 - \lambda + \frac{\delta}{\tilde{\lambda}}\right) \alpha R^k + \frac{\delta}{\tilde{\lambda}} \alpha (I - R)^{-1} \left(R^{k-m+1} - R^k\right),$$

with $R = -\tilde{\lambda}(S - \tilde{\lambda} I + \tilde{\lambda} 1_{n_p} \alpha)^{-1}$.

4.3 Exponential job sizes

If we further restrict to exponential job sizes, Theorem 8 further simplifies as $\alpha = 1$ and $R = \tilde{\lambda}$. In such case we can also analytically invert the Laplace transform of the response time distribution given by (8).

Theorem 9 *Suppose G is exponentially distributed with mean 1. The pdf of the response time distribution is given by*

$$\begin{aligned} f_R(t) = \frac{e^{-t(1-\tilde{\lambda})}}{\lambda} \left(\frac{\delta}{\tilde{\lambda}^{m-1}(1-\tilde{\lambda})} - \frac{\delta \tilde{\lambda}}{1 - \tilde{\lambda}} + \tilde{\lambda}(1-\lambda) \right) \\ - \frac{\delta e^{-t}}{\lambda(1-\tilde{\lambda})} \sum_{k=1}^{m-1} \frac{t^{k-1}(1-\tilde{\lambda}^{m-k})}{(k-1)! \tilde{\lambda}^{m-k}}. \end{aligned} \tag{18}$$

Proof For exponential job sizes with mean 1 we have $Y^*(s) = G^*(s) = 1/(1+s)$, where the first equality follows from the memorylessness. After some simplifications, we further get

$$\xi(G^*(s)) = \frac{(1-\tilde{\lambda})(1+s)}{1 - \tilde{\lambda} + s}.$$

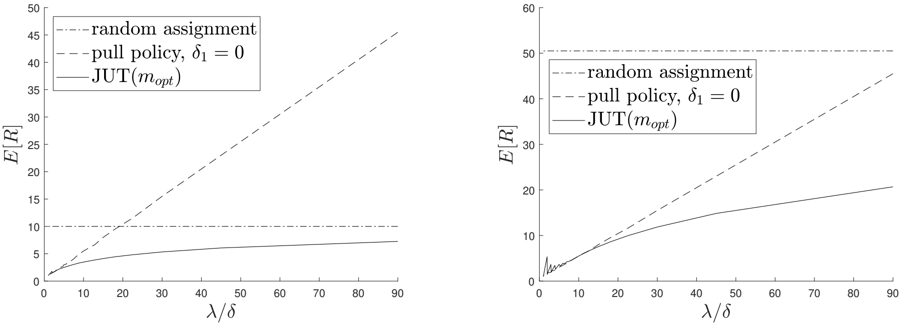

Fig. 2 Mean response time of JUT(m_{opt}), random assignment and the pull policy of [7] for $\lambda = 0.9$ and exponential (top) or hyperexponential (bottom) job sizes.

Then, by using (5), the equation above and $\lambda(1-\tilde{\lambda})/(\delta_0 m + \tilde{\lambda}) = 1 - \lambda$, we have

$$\pi(G^*(s)) = \frac{(1-\lambda)(1+s)}{1-\tilde{\lambda}+s} + \frac{\delta}{1-\tilde{\lambda}+s}\sum_{i=0}^{m-1}\left(\frac{1}{1+s}\right)^i.$$

By putting everything together, it follows that

$$\begin{aligned} R^*(s) &= \frac{\tilde{\lambda}(1-\lambda)}{\lambda(1-\tilde{\lambda}+s)} + \frac{\delta}{\lambda}\left(\frac{\tilde{\lambda}}{1-\tilde{\lambda}+s}+1\right)\sum_{i=1}^{m}\left(\frac{1}{1+s}\right)^i \\ &= \frac{1}{\lambda(1-\tilde{\lambda}+s)}\left(\tilde{\lambda}(1-\lambda) + \delta\sum_{i=0}^{m-1}\left(\frac{1}{1+s}\right)^i\right). \end{aligned}$$

Applying the inverse Laplace transform to $R^*(s)$ gives

$$\begin{aligned} f_R(t) = &\frac{e^{-t(1-\tilde{\lambda})}}{\lambda\tilde{\lambda}^{m-1}}\left(\delta\sum_{i=0}^{m-1}\tilde{\lambda}^i - \tilde{\lambda}^m(1-\lambda)\right) \\ &- \frac{\delta e^{-t}}{\lambda}\sum_{k=1}^{m-1}\frac{t^{k-1}}{(k-1)!\tilde{\lambda}^{m-k}}\sum_{i=0}^{m-k-1}\tilde{\lambda}^i, \end{aligned}$$

which equals (18). □

5 Numerical Experiments

In this section we compare the performance of JUT(m) with some existing policies and look at how sensitive its performance is with respect to the parameter m. We mainly focus on the regime where the communication overhead is well below 1 message per job as simple policies otherwise exist that can achieve vanishing delays in the large-scale limit [1, 2].

First we compare the performance of JUT(m) with random assignment and with the pull policy of [7]. We did not include a comparison with the asynchronous push policy in [6] as this policy is inferior to the pull policy of [7] as illustrated in Table 1. For the pull policy in [7] we set $\delta_1 = 0$, meaning only idle servers send updates, as this tends to yield the best performance. In Figure 2 we compare the mean response time of the different policies as a function of

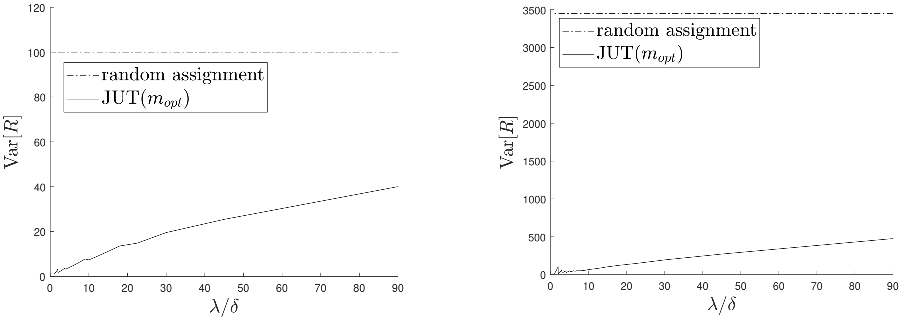

Fig. 3 Variance of the response time of JUT(m_{opt}) and random assignment for $\lambda = 0.9$ and exponential (top) or more variable (bottom) job sizes with $E[G^2] = 11$ and $E[G^3] = 330$.

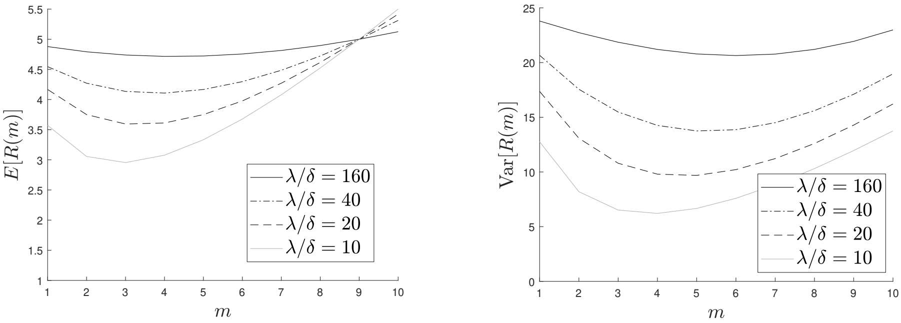

Fig. 4 Mean and variance of the response time of JUT(m) as a function of m for $\lambda = 0.8$ and exponential job sizes.

λ/δ with $\lambda = 0.9$, where δ/λ represents the mean number of communication overhead messages used per job. As random assignment does not require any communication overhead, its mean response time is fixed. We consider both exponential job sizes (in the top plot) and more variable job sizes (in the bottom plot). For the more variable job sizes we used hyperexponential job sizes with balanced means such that the squared coefficient of variation (SCV) equals 10. This implies that $E[G^2] = 11$ as $E[G^2] = SCV + 1$ (when $E[G] = 1$). Note that the mean response time of JUT(m) and random assignment only depends on $E[G^2]$ (as $E[G] = 1$), thus the results apply to any job size distribution for which the SCV equals 10.

The results in Figure 2 clearly show that the mean response time of the pull policy grows almost linearly in λ/δ and therefore the pull policy only outperforms random assignment when λ/δ is small enough, meaning when the communication overhead is large enough. The mean response time of the JUT(m) policy with $m = m_{opt}$ on the other hand grows much more slowly, is superior to random assignment for any λ/δ and outperforms the pull policy unless λ/δ is close to one. We further note that both the pull and JUT(m) policy perform better compared to random assignment when the job sizes are more variable.

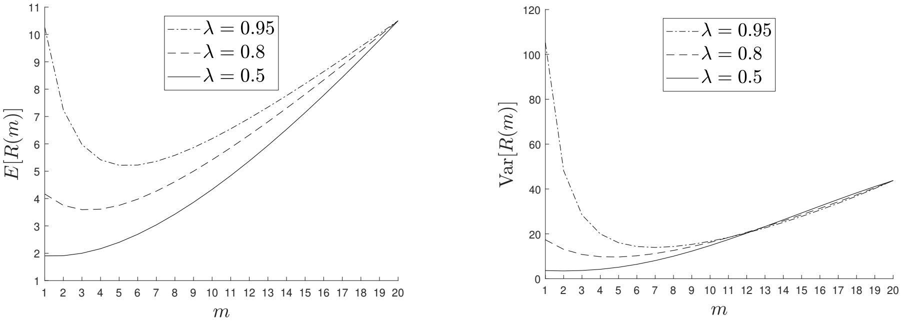

Fig. 5 Mean and variance of the response time of JUT(m) as a function of m for $\lambda/\delta = 20$ and exponential job sizes.

The previous results focused on the mean response time. We now consider the variance of the response time. As no results for the variance of the response time where presented in [7] for the pull policy, we only compare the variance of the response time of JUT(m) with random assignment in Figure 3. We consider the same two job size distributions as in Figure 2 and again set $\lambda = 0.9$. Note that for JUT(m) and random assignment the variance is only affected by the first three moments of the job size distribution. We see that JUT(m_{opt}) not only outperforms random assignment in terms of the mean response time for any λ/δ, but also significantly reduces the variance in all cases.

In the previous experiments we set $m = m_{opt}$, we now look at the impact of m on the mean and variance of the response time of JUT(m). We start by assuming exponential job sizes and set $\lambda = 0.8$. In Figure 4 we consider $\lambda/\delta \in \{10, 20, 40, 160\}$. We note that the mean response time is not highly sensitive to the choice of m, especially when the communication overhead is small (that is, λ/δ is large). This means that it suffices to get a good estimate of the arrival rate λ and the first two moments $E[G]$ and $E[G^2]$ of the job size distribution to get near optimal performance as m_{opt} does not depend on any other job size characteristics. We further note that while the value of m that minimizes the mean response time does not minimize the variance of the response time, it does yield a near optimal variance. The value of m that actually minimizes the variance appears to be somewhat larger than the value of m that minimizes the mean.

In Figure 5 we consider the same scenario as in Figure 4, but now we fix $\lambda/\delta = 20$ and let $\lambda \in \{0.5, 0.8, 0.95\}$. The results indicate that the choice of m appears to become more important as λ increases (for instance simply setting $m = 1$ is far from optimal for larger λ). The mean response time of random assignment (which requires no overhead) equals $1/(1-\lambda)$. Hence, the JUT(m) policy mostly offers a significant reduction over random assignment when λ is large. Regarding the variance of the response time, we can make the same remarks as in Figure 4.

In the previous two figures jobs were assumed to have an exponentially distributed size. We now consider job size distributions that are more variable.

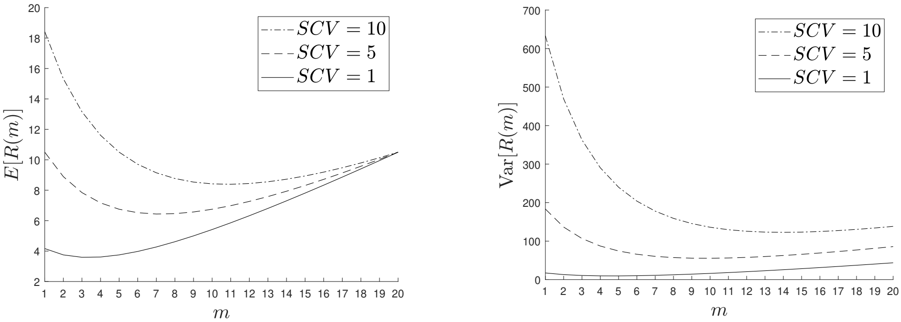

Fig. 6 Mean and variance of the response time of JUT(m) as a function of m for $\lambda/\delta = 20$ and $\lambda = 0.8$.

As before we consider hyperexponential jobs sizes with balanced means such that the squared coefficient of variation (SCV) equals $1, 5$ and 10. This implies that $E[G^2] \in \{2, 6, 11\}$ and $E[G^3] \in \{6, 90, 330\}$. Figure 6 depicts the results for $\lambda/\delta = 20$ and $\lambda = 0.8$. Similar trends are observed for the three SCV values considered. We further note that the mean response time becomes insensitive to the job size distribution when $m = \lambda/\delta$ as the mean response time reduces to $1 + (\lambda/\delta - 1)/2$ in such case according to (6).

6 Conclusion

In this paper we introduced a novel hyper-scalable load balancing policy, called JUT(m), where m is an input parameter. We studied the performance of the JUT(m) policy in a large-scale system using the queue at the cavity approach and demonstrated the accuracy of this approach using simulation.

Closed form results were presented for the generating function of the queue length distribution and the Laplace transform of the response time distribution. Using these results we derived a simple closed form solution for the mean response time and the value of m that minimizes the mean response time. Numerical results illustrate that the JUT(m) policy is superior to existing policies when the communication overhead is well below one message per job and outperforms random assignment irrespective of the communication overhead allowed. The performance gain achieved also increases as the job sizes become more variable.

Convergence towards the queue at the cavity for exponential job sizes and bounded queues was discussed in Appendix A, where the remaining technical challenges were outlined to prove weak convergence of the stationary measures of the stochastic systems as the number of servers N tends to infinity.

References

[1] Lu, Y., Xie, Q., Kliot, G., Geller, A., Larus, J.R., Greenberg, A.: Join-Idle-Queue: A novel load balancing algorithm for dynamically scalable

web services. Perform. Eval. **68**, 1056–1071 (2011). https://doi.org/10.1016/j.peva.2011.07.015

[2] Stolyar, A.L.: Pull-based load distribution in large-scale heterogeneous service systems. Queueing Systems **80**(4), 341–361 (2015). https://doi.org/10.1007/s11134-015-9448-8

[3] Anselmi, J.: Combining Size-Based Load Balancing with Round-Robin for Scalable Low Latency. IEEE Transactions on Parallel and Distributed Systems **31**(4), 886–896 (2020). https://doi.org/10.1109/TPDS.2019.2950621

[4] van der Boor, M., Zubeldia, M., Borst, S.: Zero-wait load balancing with sparse messaging. Operations Research Letters **48**(3), 368–375 (2020). https://doi.org/10.1016/j.orl.2020.04.006

[5] Gamarnik, D., Tsitsiklis, J.N., Zubeldia, M.: Delay, memory, and messaging tradeoffs in distributed service systems. ACM SIGMETRICS Performance Evaluation Review **44**(1), 1–12 (2016). https://doi.org/10.1287/stsy.2017.0008

[6] van der Boor, M., Borst, S., van Leeuwaarden, J.: Hyper-scalable JSQ with sparse feedback. Proceedings of the ACM on Measurement and Analysis of Computing Systems **3**(1), 1–37 (2019). https://doi.org/10.1145/3322205.3311075

[7] Hellemans, T., Kielanski, G., Van Houdt, B.: Performance of load balancers with bounded maximum queue length in case of non-exponential job sizes. IEEE/ACM Transactions on Networking (2022). https://doi.org/10.1109/TNET.2022.3221283

[8] Bramson, M., Lu, Y., Prabhakar, B.: Randomized load balancing with general service time distributions. In: ACM SIGMETRICS 2010, pp. 275–286 (2010). https://doi.org/10.1145/1811039.1811071. http://doi.acm.org/10.1145/1811039.1811071

[9] Shneer, S., Stolyar, A.: Large-scale parallel server system with multi-component jobs. Queueing Systems **98**, 21–48 (2021). https://doi.org/10.1007/s11134-021-09686-y

[10] Anselmi, J., Dufour, F.: Power-of-d-choices with memory: Fluid limit and optimality. Mathematics of Operations Research **45**(3), 862–888 (2020). https://doi.org/10.1287/moor.2019.1014

[11] Hellemans, T., Van Houdt, B.: On the Power-of-d-choices with Least Loaded Server Selection. Proc. ACM Meas. Anal. Comput. Syst. **2**(2) (2018). https://doi.org/10.1145/3224422

[12] Gast, N.: Expected values estimated via mean-field approximation are 1/N-accurate. Proceedings of the ACM on Measurement and Analysis of Computing Systems **1**(1), 17 (2017). https://doi.org/10.1145/3084454

[13] Gross, D., Shortle, J.F., Thompson, J.M., Harris, C.M.: Fundamentals of Queueing Theory, 4th edn. Wiley-Interscience, USA (2008). https://doi.org/10.5555/1972549

[14] Fuhrmann, S.W.: A Note on the M/G/1 Queue with Server Vacations. Operations Research **32**(6), 1368–1373 (1984)

[15] Fuhrmann, S.W., Cooper, R.B.: Stochastic Decompositions in the M/G/1 Queue with Generalized Vacations. Operations Research **33**(5), 1117–1129 (1985)

[16] Neuts, M.F.: Matrix-geometric Solutions in Stochastic Models: an Algorithmic Approach. John Hopkins University Press, Baltimore, MD (1981)

[17] Kurtz, T.G.: Approximation of Population Processes. Approximation of Population Processes, vol. nrs. 36-40. SIAM, ??? (1981). https://books.google.be/books?id=XbDd8SIYzFYC

[18] Gast, N., Gaujal, B.: Markov chains with discontinuous drifts have differential inclusion limits. Perform. Eval. **69**(12), 623–642 (2012)

[19] Cohen, J.W.: The Single Server Queue, 2 sub edn. North-Holland Series in Applied Mathematics and Mechanics 8. North-Holland, Amsterdam,The Netherlands (1982)

A On the convergence to the queue at the cavity

In this appendix we make a number of observations related to the convergence to the queue at the cavity for exponential job sizes and finite buffers of size B. The results in this section can also be generalized to phase-type distributed job sizes (which complicates notations). In (1*) we show that the stochastic system consisting of N servers is a density dependent population process as defined by Kurtz [17]. In (2*) we present an expression for the drift function, which is not continuous everywhere, and define a differential inclusions based on the drift function. Leveraging the framework in [18] allows us to show that the sample paths of the stochastic systems converge to the set of solutions of the differential inclusion over finite time scales as N tends to infinity. If the differential inclusion has multiple solutions, the system may converge to any solution of the DI, depending on its random innovations. In (3*) we argue that there exists a solution of the differential inclusion that makes a so-called *sliding motion* in a certain region of the state space and this region contains a fixed point that corresponds to our queue at the cavity.

Given the above three results, weak convergence of the steady state measures to the Dirac measure of the fixed point in (3*) follows due to [18, Section 4.2] provided that we can show that the trajectory of any solution of the differential inclusion converges to this fixed point. In other words it suffices to show that the fixed point mentioned in (3*) is a global attractor of any solution to the differential inclusion in order to exchange the limits of t and N. This proof of global attraction is still missing. We comment on a possible approach at the end of this section.

(1*) We first show that the stochastic system consisting of N servers with exponential service times and finite buffers of size B is a density dependent population process. Define the variables $Y_{i,j}^{(N)}(t)$, for $0 \leq j \leq i \leq B$ as the fraction of the N servers that have queue length j and for which the dispatcher has an estimated queue length equal to $i \geq j$ at time t. Clearly, due to the exponential job sizes the variables $Y_{i,j}^{(N)}(t)$ form a continuous time Markov chain on the state space $\mathcal{S}^{(N)} = \{y_{i,j} \mid 0 \leq j \leq i \leq B, \sum_{i,j} y_{i,j} = 1, Ny_{i,j} \in \mathbb{N}\} \subseteq \mathbb{Z}^{(B+1)(B+2)/2}/N$. This Markov chain is a density dependent population process if there exists a finite set $\mathcal{L} \subset \mathbb{Z}^{(B+1)(B+2)/2}$ (with $0 \notin \mathcal{L}$), such that for each $\ell \in \mathcal{L}$ and $y \in S^{(N)}$, the rate of transition from y to $y + \ell/N$ is of the form $N\beta_\ell(y) \geq 0$, where $\beta_\ell(\cdot)$ does not depend on N. Let $e_{(i,j)} \in S^{(N)}$ be the vector with $y_{i,j} = 1$ (and zeros elsewhere). For the JUT(m) system we have three types of transitions. (1) We can have an arrival that is assigned to a queue with length j and estimated length i. These transitions are denoted as $\ell_{a(i,j)} = -e_{(i,j)} + e_{(i+1,j+1)}$ (for $i < B$) as they change the queue state from (i, j) to $(i+1, j+1)$. Let $\kappa(y)$ be the minimum of m and the smallest estimated queue length when the system is in state y, that is, $\kappa(y) = \min(m, \min\{i \mid \exists j : y_{i,j} > 0\})$. As the job arrivals occur at rate λN and a job is assigned to a queue with the smallest estimated queue length if $\kappa(y) < m$ and at random otherwise, we have

$$\beta_{\ell_{a(i,j)}}(y) = \begin{cases} 0 & i \neq \kappa(y) < m \\ \lambda y_{i,j} / \sum_s y_{i,s} & i = \kappa(y) < m \\ \lambda y_{i,j} & \kappa(y) = m \end{cases}$$

(2) A service completion can occur in a server with length j and estimated queue length i. We denote these transitions as $\ell_{s(i,j)} = -e_{(i,j)} + e_{(i,j-1)}$ for $i \geq j > 0$. As service completions do not depend on other queues we have $\beta_{\ell_{s(i,j)}}(y) = y_{i,j}$ due to the exponential service times with mean 1. (3) The last type of transition that can occur is an update from an idle server, which changes its state from $(i, 0)$ to $(0, 0)$ for $i > 0$. We denote these as $\ell_{u(i,0)} = -e_{(i,0)} + e_{(0,0)}$. As such updates occur at rate δ_0 in any idle queue, we have $\beta_{\ell_{u(i,0)}}(y) = \delta_0 y_{i,0}$. The functions $\beta_\ell(\cdot)$ do not depend on N, therefore the Markov chain is a density dependent population process.

(2*) The drift function $f(y)$, with components $f_{(i,j)}(y)$ in our case, of a density dependent population process are defined as $f(y) = \sum_{\ell \in \mathcal{L}} \beta_\ell(y)\ell$. Let $u_i = \sum_{j=0}^{i} y_{i,j}$. Given the above discussion on the transitions in $\mathcal{L}$, we have

$$\begin{aligned}f_{(i,j)}(y) &= -1[j > 0]y_{i,j} + 1[i > j]y_{i,j+1}\\ &\quad - 1[0 = j < i]\delta_0 y_{i,0} + \delta_0 1[i = j = 0]\sum_{s>0} y_{s,0}\\ &\quad - 1[i < m]\frac{\lambda}{u_i} y_{i,j} 1[\kappa(y) = i] + 1[0 < j \leq i \leq m]\frac{\lambda}{u_{i-1}} y_{i-1,j-1} 1[\kappa(y) = i-1]\\ &\quad - 1[i \geq m]\lambda y_{i,j} 1[\kappa(y) = m] + 1[i > m, j > 0]\lambda y_{i-1,j-1} 1[\kappa(y) = m],\end{aligned} \tag{19}$$

where $1[A] = 1$ if A is true and $1[A] = 0$ otherwise. The first two terms are due to the service completions, the next two due to the updates and the remaining ones are a result of the arrivals. Note that the $1[i \geq m]$ and $1[i > m]$ conditions on the last two terms can be dropped as $y_{i,j} = 0$ for $i < m$ when $\kappa(y) = m$. Further, for ease of presentation the changes needed due to having a finite B are omitted.

When the drift function $f(y)$ is Lipschitz continuous Kurtz showed that the sample paths of the stochastic system converge to the solution of the set of ODEs given by $dy(t)/dt = f(y(t))$ over any finite time interval $[0, T]$. In our case the drift function f is clearly not continuous due to the presence of the $\kappa(\cdot)$ function. The result of Kurtz was however generalized in [18, Theorem 5] to systems with drifts that contain discontinuities. More specifically, define the differential inclusion (DI) $dy(t)/dt \in F(y(t))$ with $y(0) = y_0$ where $F(y)$ is the convex closure of the set of all $f(y)$ values that can be obtained as $f(y) = \lim_n f(y_n)$ with $\lim_n y_n = y$. Let $\mathcal{G}_T(y_0)$ be the set of solutions to the DI on $[0, T]$ with $y(0) = y_0$, where a solution is an absolutely continuous function y such that $df(y)/dt \in F(y(t))$ almost everywhere. [18, Theorem 5] then implies that

$$\inf_{y \in \mathcal{G}_T(y_0)} \sup_{t \in [0,T]} \|Y^{(N)}(t) - y(t)\| \to 0,$$

in probability provided that $\sup_y \sum_{\ell \in \mathcal{L}} \beta_\ell(y) < \infty$ and $\sum_{\ell \in \mathcal{L}} \|\ell\| \sup_y \beta_\ell(y) < \infty$. Both conditions hold in our case as $\mathcal{L}$ is finite and $\sup_y \beta_\ell(y) \leq \max(1, \delta_0)$.

To define the set valued function $F(y)$, we introduce the vectors $w^k(y)$ for $k = 0, \ldots, \kappa(y) - 1$ with (i,j)-th component given by:

$$\begin{aligned}w^k_{(i,j)}(y) &= -1[j > 0]y_{i,j} + 1[i > j]y_{i,j+1}\\ &\quad - 1[i > j = 0]\delta_0 y_{i,0} + \delta_0 1[i = j = 0]\sum_{s>0} y_{s,0}\\ &\quad - \lambda 1[k = i] + \lambda 1[j > 0, k = i-1].\end{aligned} \tag{20}$$

Looking at (19) one finds that the set $F(y)$ is defined as the convex closure of the set $\{w^0(y), \dots, w^{\kappa(y)-1}(y), f(y)\}$. When $\kappa(y) = m$ this means that $F(y)$ contains all functions $\tilde{f}(y)$, with components $\tilde{f}_{(i,j)}(y)$ of the form

$$\begin{aligned}\tilde{f}_{(i,j)}(y) = &- 1[j > 0]y_{i,j} + 1[i > j]y_{i,j+1}\\ &- 1[i > j = 0]\delta_0 y_{i,0} + \delta_0 1[i = j = 0]\sum_{s>0} y_{s,0}\\ &- 1[i < m]\lambda\alpha_i + 1[0 < j \le i \le m]\lambda\alpha_{i-1}\\ &- \lambda y_{i,j}\alpha_m + 1[j > 0]\lambda y_{i-1,j-1}\alpha_m, \end{aligned} \tag{21}$$

with $\alpha_i \in [0, 1]$ and $\sum_{i=0}^m \alpha_i = 1$.

(3*) Suppose now that we are in a region of the state space where $\kappa(y) = m$ and $q_0(y) = \sum_{i\ge0} y_{i,0} \le 1 - \lambda$. In order to remain in this part of the state space by making a so-called sliding motion, the drift of $y_{0,0}$ should be zero, such that $y_{0,0}$ remains zero. By (21) and the fact that $y_{i,j} = 0$ for $i < m$ when $\kappa(y) = m$ shows that $\alpha_0 = \delta_0 q_0(y)/\lambda$. Further, if we demand that $y_{i,j}$ remains zero for $0 < i < m$, then (21) indicates that $\alpha_i = \alpha_{i-1}$. As the sum of all α's equals one, we have $\alpha_m = 1 - \delta_0 q_0(y)m/\lambda \le 1$ when $q_0(y) \le 1 - \lambda$ due to our assumption throughout the paper that $\lambda > \delta m$ and the fact that $\delta_0 = \delta/(1-\lambda)$. If we now focus on the region with $q_0(y) = 1 - \lambda$ during this sliding motion, we find that $\alpha_i = \delta/\lambda$ for $0 < i < m$ and $\alpha_m = 1 - \delta m/\lambda = \tilde{\lambda}/\lambda$. When we plug in the above α values in (21), we find

$$\begin{aligned}\tilde{f}_{(i,j)}(y) = &- 1[j > 0]y_{i,j} + 1[i > j]y_{i,j+1}\\ &- 1[i > j = 0]\delta_0 y_{i,0} + \delta_0 1[i = j = 0]\sum_{s>0} y_{s,0}\\ &- 1[i = j < m]\delta + 1[0 < j = i \le m]\delta\\ &- \tilde{\lambda} y_{i,j} + 1[j > 0]\tilde{\lambda} y_{i-1,j-1},\\ = &- 1[j > 0]y_{i,j} + 1[i > j]y_{i,j+1}\\ &- 1[i > j = 0]\delta_0 y_{i,0} + 1[i = j = m]\delta_0(1 - \lambda)\\ &- \tilde{\lambda} y_{i,j} + 1[j > 0]\tilde{\lambda} y_{i-1,j-1}, \end{aligned} \tag{22}$$

where the second equality is due to $\sum_{s>0} y_{s,0} = \sum_{s\ge m} y_{s,0} = 1 - \lambda = \delta/\delta_0$ when $\kappa(y) = m$.

If we sum these drifts over i and use the fact that $\sum_{i>0} y_{i,0} = 1-\lambda$, we find

$$\begin{aligned}\sum_{i\ge j} \tilde{f}_{(i,j)}(y) = &- 1[j > 0]y_j + y_{j+1} - 1[j = 0]\delta_0(1 - \lambda)\\ &+ 1[j = m]\delta_0(1 - \lambda) - \tilde{\lambda} y_j + 1[j > 0]\tilde{\lambda} y_{j-1}, \end{aligned} \tag{23}$$

where $y_j = \sum_{i\ge j} y_{i,j}$. Recall now that the queue at the cavity for the JUT(m) policy with exponential job sizes is defined as an M/M/1 queue with arrival rate $\tilde{\lambda} = \lambda - \delta m$, except that when the queue is empty there are also batch arrivals of size m that occur at rate $\delta_0 = \delta/(1-\lambda)$ such that the probability that the queue is idle is given by $1-\lambda$. By demanding that $\sum_{i \geq j} \tilde{f}_{(i,j)}(y) = 0$ and by replacing $1-\lambda$ by y_0, we obtain the balance equations of such an M/M/1 queue given by

$$\begin{aligned} y_0(\delta_0 + \tilde{\lambda}) &= y_1, \\ y_m(1+\tilde{\lambda}) &= \tilde{\lambda} y_{m-1} + y_{m+1} + y_0 \delta_0 \\ y_j(1+\tilde{\lambda}) &= y_{j+1} + \tilde{\lambda} y_{j-1}, \end{aligned}$$

for $m \neq j > 0$.

This completes items (1*) to (3*). To prove convergence of the stationary measures, we must show that the fixed point of (3*) is a global attractor for any solution of the differential inclusion. This could be done by first showing that there is a unique solution and subsequently showing that any trajectory of this solution (for any starting point y_0, including all points with $\kappa(y_0) < m$) converge to this fixed point. A sufficient condition such that we have at most one solution is that the set valued function $F(y)$ is *one-sided Lipschitz.* This means that for any $y, y' \in \mathbb{R}^{(B+1)(B+2)/2}$ and any $z \in F(y), z' \in F(y')$ we have

$$\langle y - y', z - z' \rangle \leq L \|y - y'\|^2,$$

for some constant L, where $\langle x, y \rangle$ is the inner product. The following example indicates that the set valued function $F(y)$ that characterizes our differential inclusion is not one-sided Lipschitz. Let $m = 1$ and let y be such that $y_{0,0} = \epsilon, y_{1,1} = 1 - \epsilon$ which implies that the only non-zero $f(y)$ components are $f_{0,0}(y) = -\lambda, f_{1,0}(y) = 1 - \epsilon$ and $f_{1,1}(y) = \lambda + \epsilon - 1$. Let y' be such that $y'_{1,1} = 1 - 2\epsilon$ and $y'_{2,1} = 2\epsilon$, then the non-zero components of $f(y')$ are given by $f_{1,0}(y') = 1 - 2\epsilon, f_{1,1}(y') = -1 + 2\epsilon - \lambda$, $f_{2,2}(y') = \lambda$, $f_{(2,0)}(y') = 2\epsilon$ and $f_{(2,1)}(y') = -2\epsilon$. As $f(y) \in F(y)$ and $\|y - y'\|^2 = O(\epsilon^2)$, we must have that $\langle y - y', f(y) - f(y') \rangle = O(\epsilon^2)$. However,

$$\begin{aligned} \langle y - y', f(y) - f(y') \rangle &= \epsilon(f_{(0,0)}(y) - f_{(0,0)}(y')) + \epsilon(f_{(1,1)}(y) - f_{(1,1)}(y')) \\ &\quad - 2\epsilon(f_{(2,1)}(y) - f_{(2,1)}(y')) \\ &= -\epsilon\lambda + \epsilon(2\lambda - \epsilon) - 2\epsilon(-2\epsilon) = \lambda\epsilon + O(\epsilon^2). \end{aligned}$$

Hence, $F(y)$ is not one-sided Lipschitz and the uniqueness of the solution of the differential inclusion must be proven in some other manner. One possible approach could be to find a change of variables such that the set valued drift does become one-sided Lipschitz. Once uniqueness of the solution is established, one can try to use monotonicity arguments to prove global attraction of the fixed point in (3*).

B Calculation of second (raw) moment of the response time

In this appendix we derive a formula for the second (raw) moment $E[R^2]$ of the response time, which combined with $E[R]$ yields a formula for the variance $Var[R]$. We have $E[R^2] = R^{*\prime\prime}(0)$. By using (8) together with $G^{*\prime}(0) = -E[G] = -1$, $-Y^{*\prime}(0) = E[Y] = E[G^2]/2$ and $G^{*\prime\prime}(0) = E[G^2]$, we obtain

$$\begin{aligned} E[R^2] = \tilde{\lambda} Y^{*\prime\prime}(0) & \\ & + \frac{\tilde{\lambda}}{\lambda}\left(\pi''(1) + 2\pi'(1)E[G^2] + (1-\lambda)E[G^2]\right) \\ & + \frac{\delta}{\lambda}\left(E[G^2]\frac{m(m+1)}{2} + \frac{(m-1)m(m+1)}{3}\right). \end{aligned}$$

One readily checks that $Y^{*\prime\prime}(0) = E[G^3]/3$ and we already know that $\pi'(1) = \lambda E[R]$. From (4) we have

$$\begin{aligned} \pi''(1) = & \frac{\lambda}{\tilde{\lambda}}\left(\beta''(1) + 2\beta'(1)\xi'(1) + \xi''(1)\right) \\ & - \frac{\delta(m-2)(m-1)m}{3\tilde{\lambda}}. \end{aligned}$$

Making use of (5) one finds

$$\beta''(1) = \frac{\delta}{\lambda(1-\tilde{\lambda})}\sum_{i=1}^{m-1} i(i-1) = \frac{\delta(m-2)(m-1)m}{3\lambda(1-\tilde{\lambda})}.$$

We still need to find $\xi''(1)$. Denote respectively by $\tilde{R}$ and W the response and waiting time of an ordinary $M/G/1$ queue with arrival rate $\tilde{\lambda}$. By using [13, (5.30)], we get $\xi''(1) = \tilde{\lambda}^2 E[\tilde{R}^2]$. We have $E[\tilde{R}^2] = E[(W+G)^2] = E[W^2] + 2E[W]E[G] + E[G^2]$. As $E[G] = 1$ and $E[W] = \frac{\tilde{\lambda}E[G^2]}{2(1-\tilde{\lambda})}$, we obtain $2E[W]E[G] + E[G^2] = E[G^2]/(1-\tilde{\lambda})$. $E[W^2]$ is given by [19, p.256]:

$$E[W^2] = \frac{\tilde{\lambda}^2 E[G^2]^2}{2(1-\tilde{\lambda})^2} + \frac{\tilde{\lambda}E[G^3]}{3(1-\tilde{\lambda})}.$$

It follows that

$$\xi''(1) = \frac{\tilde{\lambda}^4 E[G^2]^2}{2(1-\tilde{\lambda})^2} + \frac{\tilde{\lambda}^3 E[G^3]}{3(1-\tilde{\lambda})} + \frac{\tilde{\lambda}^2 E[G^2]}{1-\tilde{\lambda}}.$$

Putting everything together, we get

$$\begin{aligned}
E[R^2] = {} & \frac{\tilde{\lambda}E[G^3]}{3} + \frac{\delta\tilde{\lambda}(m-2)(m-1)m}{3\lambda(1-\tilde{\lambda})} \\
& + \tilde{\lambda}\frac{\delta m(m-1)}{\lambda(1-\tilde{\lambda})}\left(1 + \frac{\tilde{\lambda}E[G^2]}{2(1-\tilde{\lambda})}\right) \\
& + \frac{\tilde{\lambda}^4 E[G^2]^2}{2(1-\tilde{\lambda})^2} + \frac{\tilde{\lambda}^3 E[G^3]}{3(1-\tilde{\lambda})} + \frac{\tilde{\lambda}^2 E[G^2]}{1-\tilde{\lambda}} \\
& + 2\tilde{\lambda}E[G^2]\left(1 + \frac{\tilde{\lambda}E[G^2]}{2(1-\tilde{\lambda})} + \frac{\delta}{\lambda}\frac{m(m-1)}{2(1-\tilde{\lambda})}\right) \\
& + \frac{\tilde{\lambda}(1-\lambda)}{\lambda}E[G^2] \\
& + \frac{\delta}{\lambda}\left(E[G^2]\frac{m(m+1)}{2} + \frac{(m-1)m(m+1)}{3}\right).
\end{aligned}$$